Grenze International Journal of Engineering and Technology, June Issue

"Classroom Guardian"-Detecting Unusual Activities in Classroom using Data Augmentation

Neelam Chandolikar[1], Sarthak Bhoknal[2], Nirbhay Bhirangi[3], Shrirang Bhende[4], Nikunj Bhattatd[5], Karnsingh Bhosale[6] and Prathamesh Bhosale[7]
[1-7]Department of Engineering, Sciences and Humanities (DESH), Vishwakarma Institute of Technology, Pune, Maharashtra, India
Email: neelam.chandolikar@vit.edu, sarthak.bhoknal23@vit.edu, nirbhay.bhirangi23@vit.edu, shrirang.bhende23@vit.edu, nikunj.bhattad23@vit.edu, Karnsingh.bhosale23@vit.edu, Prathamesh.bhosale23@vit.edu

***Abstract*—This project, introduces a user-friendly and efficient system to detect unusual activities in classrooms. By leveraging Python, YOLOv8, OpenCV, Face recognition, Haar Cascade classifiers, NumPy, and PyAudio, our approach covers five essential aspects: fight detection, fire detection, crowd detection, fall detection, noise monitoring. The crowd detection feature analyzes the density of people in the classroom, while fall detection identifies sudden posture changes for prompt response to potential accidents. The noise detection module categorizes unusual sounds, contributing to a safer learning environment. The fire and fight detection prevents happening of any emergency. This practical and adaptable solution enhances security and well-being in educational settings, offering a versatile tool for real-time monitoring and addressing diverse classroom scenarios. The modular codebase ensures scalability and the potential for future enhancements, emphasizing the project's applicability and longevity.**

***Index Terms*— (Unusual activity detection, Python, YOLOv8, OpenCV, Haar Cascade, PyAudio, fight, fire, crowd and fall detection, noise monitoring)**

I. INTRODUCTION

The " Detecting Unusual Activities in classroom using Data Augmentation " project represents a pioneering effort to bolster safety measures within educational environments. Employing a sophisticated combination of Python, YOLOv8[1][2][3], OpenCV, Haar Cascade classifiers, NumPy, and PyAudio, the system is designed to detect and address unusual activities comprehensively. The project integrates cutting-edge technologies to create a versatile and user-friendly monitoring solution. By responding promptly to potential incidents, the project aims to establish a secure and conducive learning environment. This innovative initiative not only emphasizes real-time security but also boasts a modular codebase, allowing for scalability and adaptability. The system's practical application in diverse educational settings positions it as a valuable tool, promising both immediate impact and long-term efficacy. The project stands at the forefront of technological advancements, offering a robust solution to the evolving security needs of modern classrooms.

II. LITERATURE REVIEW

Talaat et.al [1] The research paper introduces an enhanced fire detection approach based on YOLO-v8 for smart

Grenze ID: 01.GIJET.10.2.407

cities. The focus is likely on utilizing the advanced capabilities of YOLO-v8 to improve the accuracy and speed of fire detection.The proposed system is expected to contribute to early and efficient fire detection, aiding in rapid response

Long et.al [2] The research paper focuses on an image-based fall detection system employing (YOLO) algorithm to monitor falls among elderly individuals. The study likely explores how YOLO, a real-time object detection algorithm, can be utilized for rapid and accurate identification of fall events through analyzing images. The system is utilizing the efficiency of YOLO for real time monitering.

Bo, L. U. O. et.al [3] The research paper explores Human Fall Detection for Smart Home Care using YOLO networks. YOLO is a real-time object detection system. The paper likely discusses how YOLO can be applied to detect falls within a smart home environment, enhancing elderly care through automated monitoring and timely assistance.The approach of YOLO's efficiency in processing images or video frames to quickly and accurately identify human falls

Ramzan, Muhammad et.al [4] This research paper focuses on detecting and recognizing unusual activities in a classroom situation such as examination halls, which may help the invigilators observe and restrict the students from cheating or using unfair means by using a trained CNN model named (AUAR)

Senthilkumar, T. et.al [5]: This research paper aims to develop a system that analyses and detects suspicious activity that often occurs in a classroom environment. This project is done by using C/C++ and OpenCV.

Kumar, Manoj et.al [6] This research paper helps to classify the activities as normal and abnormal activity. This paper works on the CNN model.

Mr. Bibhu Ranjan Sahoo et.al [7] This research paper aims to automate the monitoring of unusual human activities such as fall detection. This project work on a model named STAE (Spatial Temporal Auto Encoder).

Kim, Eunju et.al [8] This research paper gives an idea about detecting basic human activities by using HMM model. (Hidden Morkov Model)

Yin et.al [9] This project primarily focuses on detecting unusual human activity using wireless sensors attached to human body.

B Venkata Ramana et.al [10 This research papers helps us in understanding human activity recognition by using opencv and a trained CNN model.

Sawant, Chinmay et.al [11] This project predicts the human activity using Long short term memory which is a RNN model

Chaudhari, Mayur D. et.al [12] This project predict crowd detection by using Pattern Recognition, Computer Vision, Crowd Density Estimation, Detection, CNN.

Sarveshwaran et.al [13] This paper detect classify human activity using deep learning and CNN model

III. METHODOLOGY/EXPERIMENTAL

The scope of the project is to develop a system that can detect unusual activities in a classroom setting. It detects various types of unusual activities like 1) Fight, Fire and Fall Detection 2) Crowd Detection and 3) Noise Detection. Different methods and different datasets are used for detection of above types of activities in classroom. further details are as follows.

A) Fight, Fire and Fall Detection

The scope of the project is to develop a system that can detect unusual activities in a classroom setting, specifically fire, fight, and fall. The objective is to achieve real-time detection and accurate classification of these activities. To achieve this, data acquisition is necessary, and a labeled dataset containing diverse images and videos of classroom scenarios is required. Roboflow[14] datasets [15,16,17] are used for training and model preparation for fire, fight, and fall detection.

After acquiring the dataset, the next step is to preprocess it to ensure consistency and quality. This includes resizing images to a standard resolution, normalizing pixel values, and addressing any class imbalance. Data augmentation techniques should also be applied to increase dataset diversity, such as random rotations, flips, and translations, changes in brightness, contrast, and saturation, random cropping and scaling.

The dataset should be divided into training, validation, and test sets, with 80% for training, 10% for validation, and 10% for testing. Each set should have a representative distribution of unusual activities. YOLOv8 is chosen as the base model for its real-time detection capabilities and object localization accuracy.

The YOLOv8 model is trained using the augmented dataset, initialized with pre-trained weights, and fine-tuned on the unusual activities dataset. After training, the model's performance is evaluated on the validation and test sets, calculating precision, recall, and F1 score for each class, and assessing generalization to new, unseen data.

To present alerts for detected unusual activities, we display a live video feed and deploy the system in the classroom environment. Extensive testing is conducted in real-world conditions to validate the model's accuracy in detecting unusual activities. Based on the performance analysis, we iterate on the model, data augmentation, and system parameters, leading to iterative improvement. Figure 1 shows method for Fight, Fire, Fall detection.

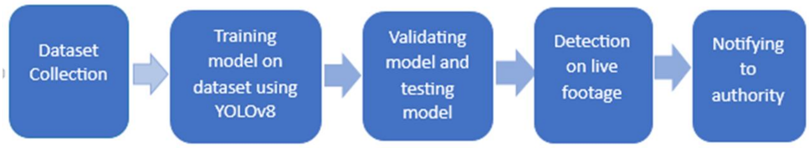

Fig. 1. Block-Diagram for Fight, Fire, Fall detection.

B) Crowd Detection

The initial step for crowd detection using OpenCV, face recognition involves importing essential libraries, including OpenCV for computer vision tasks, the time module for time-related operations, the notification module for local system notifications, and the Pushbullet library for cross-device notifications. Subsequently, the pre-trained Haar Cascade classifier for face detection is loaded to facilitate the identification of faces in the video stream. Key parameters, such as the crowd threshold, maximum face height, and notification interval, are then configured to fine-tune the detection process.

The code initializes crucial variables like 'notification_sent' to track notification status and timers for managing notification intervals. Pushbullet is set up by providing the required API key, creating a Pushbullet object, and ensuring proper connectivity for sending notifications to other devices. The core face detection function, 'detect_faces,' is designed to process each frame by converting it to grayscale, detecting faces, and filtering out those in a standing position based on maximum height.

Within the main loop, the code continually captures video frames, counts standing faces using the 'detect_faces' function, and displays the processed frame with face detection information. Upon surpassing the defined crowd threshold and ensuring that a notification has not been sent within the specified interval, the program displays a crowd detection message on the frame and sends notifications both locally using 'plyer' and remotely using Pushbullet. To prevent continuous notifications, a 10-second delay timer is initiated.Overall, this methodology outlines a comprehensive approach to real-time crowd detection, combining computer vision, notification mechanisms, and efficient resource handling. Figure 2 shows method for crowd detection module.

Fig. 2. Block-Diagram for Crowd detection.

C) Noise Detection

It integrates the PyAudio library for capturing audio input and OpenCV for accessing the webcam. The primary goal is to identify noise events based on amplitude levels in the audio stream and provide visual feedback using the webcam feed.

The code starts by setting a threshold for noise detection (THRESHOLD_AMPLITUDE) and initializing a Pushbullet instance with an API key (PUSHBULLET_API_KEY). It also establishes variables for tracking notification timestamps and defines a notification delay to prevent continuous notifications.

Two functions, send_plyer_notification and send_pushbullet_notification, handle the notification process. The former utilizes the plyer library to display a desktop notification, while the latter sends a notification to Pushbullet. Both functions incorporate a delay mechanism to control the frequency of notifications.

The detect_noise function is at the core of the noise detection process. It reads audio data from the PyAudio stream, calculates the maximum amplitude in the current frame, and checks if it exceeds the predefined threshold. Simultaneously, it captures a frame from the webcam using OpenCV. If noise is detected, the code overlays a "Noise detected!" on the video screen, sends a desktop notification, and send a Pushbullet notification.

Overall, this methodology outlines a comprehensive approach to real-time noise detection, leveraging both audio and visual information and incorporating notifications for user awareness. The code aims to strike a balance between responsiveness, notification frequency, and resource release in case of program termination. Figure 3 provides details of noise dectection module.

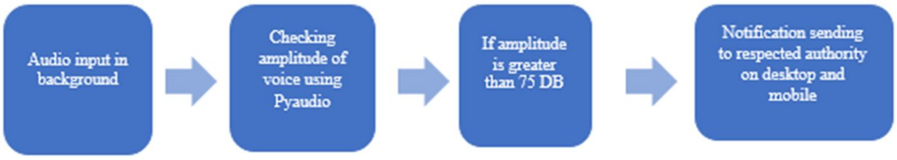

Fig. 3. Block-Diagram for Noise detection.

IV. Results And Discussions

When we first trained our YOLOv8 model for detecting fire, fights, and falls, it was giving less accuracy. Then, to improve accuracy actual pictures are taken in the classroom and the data set is augmented. his way, the model could learn about the different things in a classroom, about lighting effects, placement of different objects in classroom, size of objects etc. Using real classroom pictures during training helps the model recognize and understand unusual activities, making it work better in a classroom. Data augmentation improved accuracy.

Fig. 4. Fire Detection predicted result.

Fig. 5. Fight Detection predicted result.

Fig. 6. Fight Detection predicted result.

Fig. 7. Fall Detection predicted result.

V. Future Scope

The project on detecting unusual activities in classrooms using YOLOv8 with data augmentation holds promising future prospects. To enhance detection capabilities, future work may involve integrating advanced computer vision techniques and exploring state-of-the-art architectures. Continuous learning mechanisms should be implemented to adapt the model to evolving patterns over time, and multi-modal integration, incorporating audio and contextual information, could provide a more comprehensive understanding of activities. Behavioral analysis, focusing on patterns of movement and group dynamics, could offer deeper insights. Privacy-preserving techniques and adaptive data augmentation strategies are crucial for addressing ethical concerns and improving the model's adaptability. The integration of human pose estimation and exploration of edge computing and IoT can further enhance the system's scalability and real-time processing capabilities. Considerations for ethical deployment, robustness, user acceptance, and compliance with regulations should be central to discussions. Demonstrations and community engagement will be essential in fostering transparency and addressing concerns, ensuring that the system aligns with both security needs and ethical considerations in educational environments.

VI. CONCLUSION

In conclusion, the implementation of YOLOv8 for detecting unusual activities in a classroom setting, enriched by data augmentation techniques and leveraging a diverse dataset from Roboflow, represents a significant step towards enhancing safety and security. The YOLOv8 framework, known for its real-time object detection capabilities, serves as a robust foundation for the project. The integration of data augmentation strategies ensures that the model generalizes effectively to diverse scenarios, contributing to its reliability in real-world applications. The project's success is contingent on several factors, including the quality of the dataset, appropriate configuration of the YOLOv8 model, and the efficacy of the chosen data augmentation techniques. Continuous monitoring and updating of the model, as well as iterative improvements based on user feedback, will be essential for maintaining optimal performance over time.

Our approach provide detection of five important unusual activities fight detection, fire detection, crowd detection, fall detection, noise monitoring, in an educational classroom setup.

The future scope of the project extends towards incorporating advanced computer vision techniques, multi-modal integration, and adaptive learning mechanisms. Additionally, addressing privacy concerns, optimizing real-time performance, and exploring the integration of edge computing and IoT devices are avenues for further enhancement. Ethical considerations, community engagement, and compliance with regulations remain integral aspects for the responsible deployment of such surveillance systems in educational environments.

In summary, the YOLOv8-based unusual activity detection project stands as a proactive measure for augmenting security in classrooms, and its continued evolution and adaptation to emerging technologies will contribute to creating safer learning environments.

REFERENCES

[1] Talaat, Fatma M., and Hanaa ZainEldin. "An improved fire detection approach based on YOLO-v8 for smart cities." Neural Computing and Applications 35, no. 28 (2023): 20939-20954.

[2] Long, Koh Zhen, Hazlina Haron, Maizura Ibrahim, and Zeti Darleena Eri. "An Image-based Fall Detection System using You Only Look Once (YOLO) Algorithm to Monitor Elders' Fall Events." In Knowledge Management International Conference (KMICe) 2021, vol. 1. 2021.

[3] Bo, L. U. O. "Human Fall Detection for Smart Home Caring using Yolo Networks." International Journal of Advanced Computer Science and Applications 14, no. 4 (2023).

[4] Ramzan, Muhammad, Adnan Abid, and Shahid Mahmood Awan. "Automatic Unusual Activities Recognition Using Deep Learning in Academia." Computers, Materials & Continua 70, no. 1 (2022)

[5] Senthilkumar, T., and G. Narmatha. "Suspicious human activity detection in classroom examination." In Computational Intelligence, Cyber Security and Computational Models: Proceedings of ICC3 2015, pp. 99-108. Springer Singapore, 2016.

[6] Kumar, Manoj, Anoop Kumar Patel, Mantosh Biswas, and S. Shitharth. "Attention-based bidirectional-long short-term memory for abnormal human activity detection." Scientific Reports 13, no. 1 (2023): 14442.

[7] Kim, Eunju, Sumi Helal, and Diane Cook. "Human activity recognition and pattern discovery." IEEE pervasive computing 9, no. 1 (2009): 48-53.

[8] Yin, Jie, Qiang Yang, and Jeffrey Junfeng Pan. "Sensor-based abnormal human-activity detection." IEEE transactions on knowledge and data engineering 20, no. 8 (2008): 1082-1090.

[9] Sawant, Chinmay. "Human activity recognition with openpose and Long Short-Term Memory on real time images." EasyChair Preprint 2297 (2020).

[10] Ghotkar, A., M. D. Chaudhari, and A. S. Ghotkar. "A study on crowd detection and density analysis for safety control Indian sign language interpretation view project hand gesture recognition for HCI view project a study on crowd detection and density analysis for safety control." Artic Int J Comput Sci Eng 6 (2018).

[11] Sarveshwaran, Velliangiri, Iwin Thankumar Joseph, M. Maravarman, and P. Karthikeyan. "Investigation on Human Activity Recognition using Deep Learning." Procedia Computer Science 204 (2022): 73-80.
[12] Dwyer, B., Nelson, J. (2022), Solawetz, J., et. al. Roboflow (Version 1.0) [Software].
[13] SIGHT-Fight-Detection-Dataset.SIGHT (2023). Roboflow Universe (version1.0) [Software]. Available from https://universe.roboflow.com/sight/sight-fight-detection.
[14] fall_video Dataset. Ctuet(2023). Roboflow Universe (version1.0) [Software]. Available from https://universe.roboflow.com/ctuet/fall_video
[15] Fire Dataset. College(2024) Roboflow Universe (version1.0) [Software]. Available from https://universe.roboflow.com/college-3otff/fire-0tok9